Restructuring desktops to support prevention, detection and recovery

Jeanna Neefe Matthews
Clarkson University/VMware
jnm@clarkson.edu

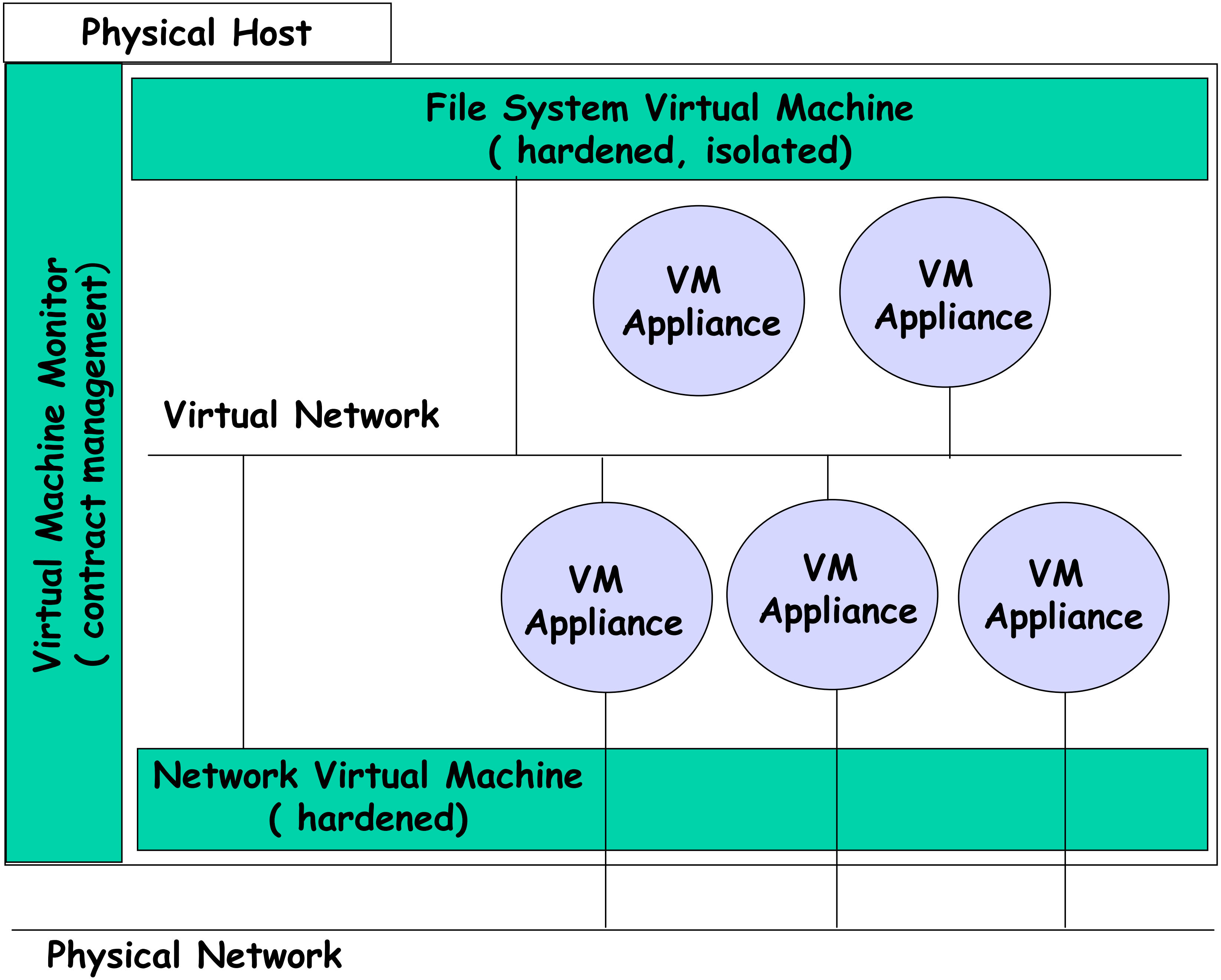
Work-in-progress presentation
LEET09, Boston MA, 04/21/09

Focus on the source?

- Studying and stopping deployed botnets is necessary, but hard!
- Should we focus also on restructuring desktops to reduce the number of 24/7 connected, nearly idle, general purpose computing resources?

Architecture to Support Prevention, Detection and Recovery

- Desktop virtualization
  - Run guest OS of choice so Windows
- Virtual machine contracts to describe expected behavior
- Firewall VM to limit network access
- File system virtual machine to protect personal data and export it to other VMs as appropriate
  - Export what is needed with only permissions needed
  - Separate system data from personal data


Physical Host
File System Virtual Machine
(hardened, isolated)
Virtual Machine Monitor
(contract management)
VM
Appliance
VM
Appliance
Virtual Network
VM
Appliance
VM
Appliance
VM
Appliance
Network Virtual Machine
(hardened)
Physical Network

Virtual Machine Contracts

- Virtual machine contracts that specify expected behavior – limit from general purpose computing device
- Examples
  - Expected rate/type of outgoing network activity
  - Open ports
  - Mount points expected into personal data store
  - Permissions on each mount point
  - Read/write rate limiting
  - Expected correlation of data access and keyboard activity
  - Resource limits
- Contracts can be inspected before running VM
  - VM's with tighter contracts that do the same job have higher value

```
<ns:ContractSection ovf:required=”true”
         xsi:type="ovf:NetworkContract_Type">
    <Info> Network Contract for Webserver </Info>
    <Rule>
       <Info> Incoming web requests </Info>
     <Protocol> tcp </Protocol>
       <DstAddr ovf:id=”webservervm” />
       <DstPort> 80 </DstPort>
       <SrcAddr> any </SrcAddr>
       <SrcPort> any </SrcPort>
    </Rule>
    <Rule>
       <Info> Connection to back-end DB </Info>
       <Protocol> tcp </Protocol>
         <DstAddr ovf:id=”dbservervm” />
       <DstPort> 3306 </DstPort>
        <SrcAddr ovf:id=”webservervm” />
       <SrcPort> any </SrcPort>
    </Rule>
</ContractSection>
```

Questions

- Who else in LEET community or related communities interested in approaches like this to attacking botnet problem at source?
- Biggest hurdles to this approach?
  - Desktop virtualization deployment? Larger deployment
  - Single-desktop experience for users? Fusion, integration with window managers
  - Hypervisor security? Not perfect but smaller/easier to harden
  - Availability of "virtual appliances"? Appliance marketplaces
  - Speed? Fast enough for typical desktop use
  - Contract standardization? Working in standards bodies like DMTF
  - OEM deployment on this configuration?
  - Will it prevent substantial category of exploits? Better approach?